Journal of Civil Structural Health Monitoring (2022) 12:127–140
https://doi.org/10.1007/s13349-021-00537-1

ORIGINAL PAPER

Two-step approach for fatigue crack detection in steel bridges using convolutional neural networks

Said Quqa[1] · Panagiotis Martakis[2] · Artur Movsessian[3] · Sai Pai[4] · Yves Reuland[2] · Eleni Chatzi[2]

Received: 20 May 2021 / Revised: 12 September 2021 / Accepted: 26 October 2021 / Published online: 5 November 2021

Abstract

The advent of parallel computing capabilities, further boosted through the exploitation of graphics processing units, has resulted in the surge of new, previously infeasible, algorithmic schemes for structural health monitoring (SHM) tasks, such as the use of convolutional neural networks (CNNs) for vision-based SHM. This work proposes a novel approach for crack recognition in digital images based on coupling of CNNs and suited image processing techniques. The proposed method is applied on a dataset comprising images of the welding joints of a long-span steel bridge, collected via high-resolution consumer-grade digital cameras. The studied dataset includes photos taken in sub-optimal light and exposure conditions, with several noise contamination sources such as handwriting scripts, varying material textures, and, in some cases, under presence of external objects. The reference pixels representing the cracks, together with the crack width and length, are available and used for training and validating the proposed model. Although the proposed framework requires some knowledge of the "damaged areas", it alleviates the need for precise labeling of the cracks in the training dataset. Validation of the model by means of application on an unlabeled image set reveals promising results in terms of accuracy and robustness to noise sources.

Keywords Vision based · Damage identification · Machine learning · Crack detection · Steel bridge · Structural health monitoring

1 Introduction

A large number of structures and significant parts of existing infrastructures built in the last century are now close to, or even beyond, the end of their design lifespan. Due to inherent defects that relate to fabrication and construction processes, as well as owing to the cyclic nature of operational (e.g., traffic) loads, a common pathology for steel infrastructures concerns the formation of fatigue cracks. Such cracks accelerate structural deterioration and, in the absence of proper inspection and maintenance, may eventually lead to failure [1].

Structural health monitoring (SHM) techniques have been proposed as a complementary practice to on-site visual inspections, which typically require substantial human resources and suffer from bias, as the inspectors' assessment is subjective [2]. SHM techniques offer a means for reducing such bias, via complementary exploitation of evidence

✉ Said Quqa
said.quqa2@unibo.it

Panagiotis Martakis
martakis@ibk.baug.ethz.ch

Artur Movsessian
artur.movsessian@ed.ac.uk

Sai Pai
sai.pai@sec.ethz.ch

Yves Reuland
reuland@ibk.baug.ethz.ch

Eleni Chatzi
chatzi@ibk.baug.ethz.ch

[1] Department of Civil, Chemical, Environmental, and Materials Engineering, University of Bologna, Viale del Risorgimento 2, 40136 Bologna, Italy

[2] Department of Civil, Environmental and Geomatic Engineering, ETH Zurich, Stefano-Franscini-Platz 5, 8093 Zurich, Switzerland

[3] School of Engineering, Institute for Infrastructure and Environment, University of Edinburgh, Alexander Graham Bell Building, Thomas Bayes Road, Edinburgh EH9 3FG, UK

[4] Singapore-ETH Centre, 1 Create Way, CREATE Tower, 138602 Singapore, Singapore

stemming from data. This evidence is built up via extraction of engineered features that are sensitive to damage and subsequent establishment of rigorous metrics which serve for damage detection and, possibly, quantification. To this end, vibration-based techniques, mostly based on ambient recordings, have gained popularity especially for bridges and slender structures [3, 4]. However, the parameters obtained using such methods generally refer to the global behavior of structures; thus, being less sensitive to local damage. In addition, information on localization may only be delivered underutilization of dense sensor networks [5]. The deployment of such dense networks on large-scale civil structures, however, leads to issues related to data management and instrumentation costs. In an attempt to relax constraints relating to cost and cabling implications, wireless sensing technologies have been adopted, requiring strict data synchronization [6–8].

When wishing to detect local defects, particularly manifestation of cracking, vision-based procedures offer a valuable enabler [9–12] due to their effectiveness in identifying structural anomalies, via use of cameras directly deployed on the structure, mounted on drones, or used by expert operators [13, 14]. Vision-based methods initially relied exclusively on image processing techniques to detect superficial defects, such as corrosion and cracks. In this context, Abdel-Qader et al. [15] offer a comparative study considering the Haar wavelet transform, the Fourier transform, and two edge detectors (namely, the Sobel and the Canny detectors) for crack identification in bridges. Edge detection methods have dominated early literature in this domain [16, 17]. However, this technique may lead to ill-posed problems, since lighting and image noise may considerably affect the analysis result, eventually yielding inaccurate crack detection [11]. Although denoising techniques may increase accuracy, their efficiency is strongly case dependent [11, 18]. A review on image processing techniques used for crack detection can be found in reference [19].

With the development of high-performance graphics processing units (GPUs) and parallel computing [20], an ever-increasing number of applications are employing machine learning and, in particular, neural networks for vision-based damage detection [21]. Convolutional neural networks (CNNs), inspired by the functioning of the visual cortex of animals, are commonly used in the vision-based field. In general, machine learning tools in vision-based applications can identify cracks at the image, block, or pixel level, according to the capability of a given algorithm to detect either the presence of a crack in an inspection image, determine the approximate cracked region, or identify the precise pixels where the crack is, respectively [22]. As an example of successful applications of machine learning in vision-based crack detection, Li and Zhao [23] modified the AlexNet convolutional neural network to classify image regions as either "cracked" or "non-cracked"; thus, achieving a block-level validation accuracy of 99.06%. Similarly, Zhang et al. [24] employed a CNN able to classify small image regions with a precision of 86.96% for road crack detection applications. Cha et al. [11] presented a study on concrete cracks, obtaining 98% accuracy on the binary classification of 256×256-pixel regions as either "damaged" or "undamaged" areas.

Recently, the pixel-level identification class has gained increasing interest, since the accurate identification of crack pixels can quantify the crack size and, thus, the entity of damage. Dung and Anh [25] employed an encoder–decoder network (called "U-net") trained end-to-end using a dataset of pixel-level annotated images to identify pixels related to reinforced concrete cracks. A similar network was employed by Huyan et al. [22] to identify cracks on road pavements with 99.01% accuracy. In most of the mentioned studies, the material background was relatively uniform and the noise sources were limited to changing lighting conditions. Xu et al. [26] used a deep fusion CNN exploiting the combination of multilevel features to achieve an overall accuracy of more than 95% in the validation phase. These results were obtained in an environment with complex disturbances given by crack-like handwriting scripts, different paint, and exposure conditions; a setting which is typically met on actual bridge systems. Other techniques [14, 27] employ machine learning-based techniques to achieve super resolution and improve crack detectability in blurred image conditions.

In further overviewing relevant machine learning-based schemes, it is worth noting the work by Dung and Ahn [25], who presented a method based on a convolutional autoencoder to directly detect cracks in photos of concrete elements. The authors also underlined the difficulty in estimating the crack size due to different noise sources present in inspection images. Further researchers attempted to characterize the identified cracks in terms of length, width, and area over the last few years. For instance, Jin et al. [28] calculated the morphological skeleton of identified crack areas and employed a flexible kernel to estimate their size.

Although much effort has been made to improve the identification accuracy, supervised machine learning approaches that provide pixel-level results require a time- and labor-intensive labeling process to build a training dataset: collect photos of damaged elements, perform manual and accurate offline labelling of the pixels corresponding to cracks, and train the CNN for further in situ identification. Also, the optimal training dataset is generally application-specific, i.e., different training sets are necessary for each case study since materials, lighting conditions, and defect types can be different. Moreover, the input of CNN-based techniques typically consists of the entire inspection image. Qiao et al. [29] used a densely connected CNN with added upsampling layers to generate output images with the same size as the input

images, which, however, involve considerable computational cost, as do the complex U-net and AlexNet architectures, largely used for vision-based applications [23, 25].

This study proposes a light crack detection tool that can be easily adapted to diverse case studies due to the low effort required to build the training set. The proposed method is devised to offer an alert of damage, together with useful quantitative information, based on photos collected manually by operators or in an automated fashion by unmanned aerial vehicles (UAVs). A central element of this study lies in minimization of false alarms triggered by different disturbance sources, such as varying light conditions and crack-similar elements (e.g., handwritten scripts). The proposed method aims at achieving a pixel-level identification by means of a hybrid algorithm based on a CNN operating at the block level and edge detection procedure, which allows identifying pixel-level cracks within the identified blocks. Compared against state-of-the-art approaches, this procedure reduces the labor related to the preparation of the training set, since the training images need to be labelled only at block level. Also, the input of the CNN is a set of small images (specific regions of the original inspection photos) that require a simpler architecture when compared against schemes in existing literature [11, 13, 22, 23, 25, 30]. Moreover, the edge detection process is applied to (small) selected regions; thus, comprising lower computational complexity than a traditional application of the same process to the entire inspection image.

Quantitative characteristics of the identified cracks (namely, total length and average width) can be identified using the proposed algorithm to provide operators with preliminary indicators of the damage entity. These quantities are expressed in pixels and can be converted to physical measurements using well-known methods, depending on the hardware employed. For instance, if the inspection camera is monocular, photogrammetry techniques can be employed to retrieve measurements from a sequence of images taken from different angles [31, 32]. Also, recent studies have demonstrated that the use of binocular cameras (generally, a regular high-resolution camera coupled with a depth sensor camera) can be effectively employed to obtain distance information from a single photo [33, 34].

The photos of fatigue cracks collected on the steel box girders of an in-service long-span bridge in China are used to validate the proposed approach. The data employed in this paper were provided in the framework of the 1st International Project Competition for Structural Health Monitoring (ICP-SHM, 2020) [35].

2 Proposed method

We propose a two-step approach (Fig. 1) to localize pixels corresponding to cracks within an image of a structural element/detail. These pixels will henceforth be referred to as "crack pixels", while we refer to the photos of structural elements as "inspection images". The first step involves identification of rectangular regions (of a fixed, user-defined size)—within the inspection image—that contain the crack. Specifically, the inspection images are divided into a grid of smaller regions without overlapping (in this study, 32×32 pixels), which are then classified as "damaged" or "undamaged" regions using a CNN that admits single regions of the inspection images as input. Image processing techniques are subsequently applied, in the second step of the procedure, only to previously identified "damaged" areas to localize the

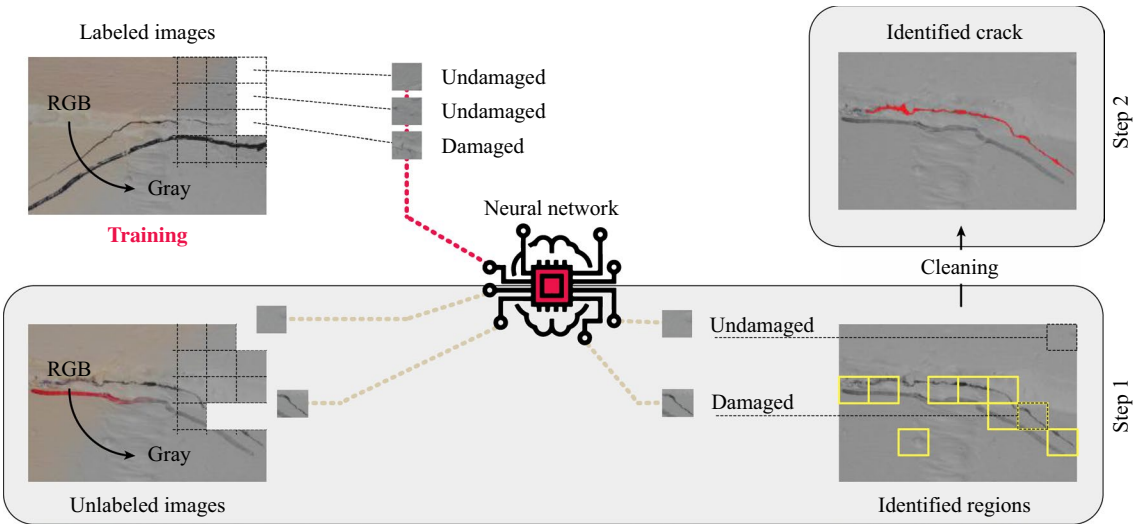


Fig. 1 Outline of the proposed procedure for crack detection

crack pixels accurately. In this way, the computational burden of image processing is minimized, and pixel-level crack identification can be easily obtained, although the training process is carried out at a block level. Depending on the nature of application and objective, Step 2 may be omitted. Moreover, between the two main steps, a cleaning procedure based on morphological operations can be applied to the outcome of the CNN to reduce false positives (FPs) and false negatives (FNs), maximizing true positives (TPs) and true negatives (TNs). Herein, "positives" denote the cracked regions, while "negatives" the pristine material, while "true" and "false" denote correctly or incorrectly identified samples, respectively.

In the next few sections, the methodology briefly described above and schematized in Fig. 1 is outlined in detail. Section 2.1 explains the pre-processing operations necessary to train the CNN. Sections 2.2 and 2.3 describe the two steps of the identification procedure that ultimately yields an estimate of the crack length and width.

2.1 Pre-processing of inspection images and training of the CNN

In general, for CNN applications, the input image size strongly affects the computational runtime both for network training and, more importantly, for the classification of new datasets. However, in most machine-learning-based procedures for crack identification, images representing large areas are directly used as input [29, 30]. In other applications, inspection images are processed using moving windows; thus, considering regions of reduced size. In particular, Cha et al. [11] used 256×256 pixel windows to identify cracked areas in reinforced concrete structures. However, when different noise sources disturb the images, the efficacy of crack identification is correspondingly hindered. The consideration of targeted localized image frames may facilitate distinguishing between actual cracked areas and crack-like noise sources. In particular, when the structural element images include handwritten scripts or when the material texture is wrinkled (e.g., welding joints), considering image regions that are a few times (i.e., 5–10 times) larger than the expected crack width can facilitate the identification of structural flaws. For this reason, in this study, the inspection images are preliminarily divided into 32×32-pixel regions that are provided as input to the proposed crack identification method. The processing of these small regions also involves faster computation, since the CNN has a 32×32 input layer and thus, requires few convolutional layers for classification. Figure 2 shows the architecture of the CNN employed in this study. Each layer is identified using an identification code (ID), which is described in Table 1.

Since Step 1 employs a CNN for region classification, it requires a preliminary training phase using labeled images, i.e., photos where the crack location is known. However, it should be noted that, contrary to existing methods, a rough knowledge of cracked areas is sufficient, considering that the labels are assigned to the 32×32 regions; this implies alleviation of the need to precisely highlight the exact crack pixels. Therefore, the training dataset consists of a set of regions labeled as "damaged" or "undamaged". This characteristic considerably simplifies the labeling process, which can be done quickly by operators using, for example, a capacitive pen on a tablet.

The inspection images are first pre-processed and divided into regions to generate a dataset suitable for training the CNN. Specifically, based on the assumption that color information in inspection images does not contribute to identifying cracks, since different lighting conditions may distort it, the original inspection images are first transformed into grayscale figures. This process further simplifies the region classification, since the greyscale input data consists of a single-color layer instead of the three-color layers of the

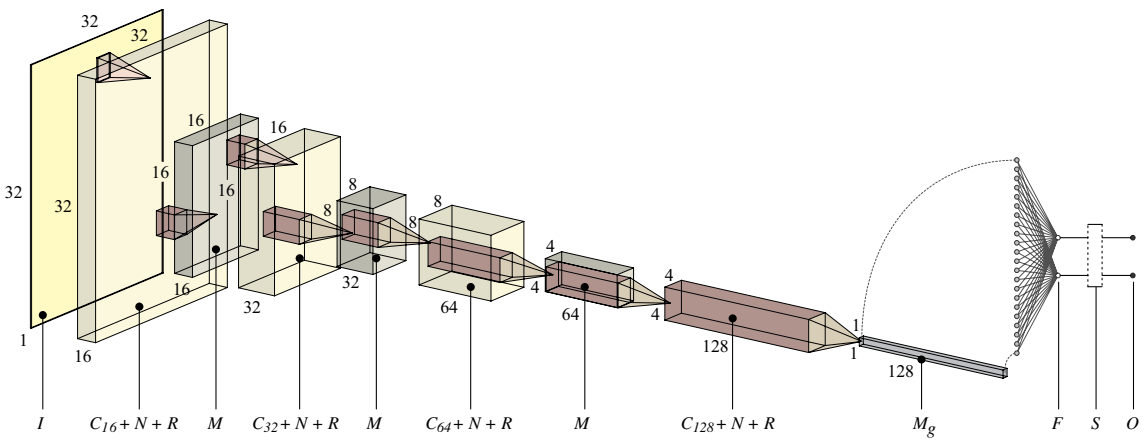


Fig. 2 Architecture of the neural network for binary classification of regions of the inspection images as "damaged" or "undamaged"

Table 1 Description of the CNN layer architecture

ID	Layer type	Details
I	Input	32×32 matrix containing the numeric values that represent the grayscale intensity of the pixels in the input region
C_s	Convolutional	s filters with 3×3 kernel; padding is selected to have an output with the same size as the input
N	Batch normalization	Reduces the effects of internal covariate shift by normalizing each input across a mini-batch
R	ReLU	Rectified linear unit layer
M	Max pooling	Performs downsampling keeping a maximum of 2×2 pooling regions, without overlapping
M_g	Global max pooling	Retains the maximum for each channel of the previous layer
F	Fully connected	Fully connected layer of two neurons
S	Softmax	Normalizes the inputs into probabilities
O	Classification	Classification layer with two outputs ("damaged" or "undamaged")

red–green–blue (RGB) coding; thus, reducing the required filters in the convolutional layers. Each grayscale image of the training set is then divided into 32×32-pixel regions forming the training samples for Step 1. It should be noted that, to increase the size of the training dataset, overlapping may be considered. However, in this study, the inspection images are divided using a regular grid without overlapping. A CNN with the structure shown in Fig. 2 is, thus, trained using the generated dataset.

2.2 Step 1: selection of the damaged regions

The trained neural network should be able to discern between damaged and undamaged 32×32 regions of the inspection image. To process a new dataset to identify the damage regions in unlabeled inspection images, a pre-processing procedure, consisting of desaturation and splitting (as described in the previous section), must be preliminarily performed. Hence, a set of testing 32×32 regions compatible with the trained CNN is generated. Each unlabeled input region can then be classified as either "damaged" or "undamaged".

At this point, a strategy based on morphological operations can be employed to mitigate FPs and FNs outcomes that may result from diverse sources of noise, as shown in Fig. 3. To this aim, a map of the classified regions in the inspection image is first generated using Boolean variables. Specifically, a matrix with dimensions $p \times q$ is built, where p and q are the number of 32×32 regions that form the inspection image in the vertical and horizontal direction, respectively. Each element of this matrix is set to either 0 or 1 (binary value), corresponding to a region being classified as "undamaged" or "damaged", respectively. A cleaning procedure is, thus, performed by converting isolated damaged segments (i.e., instances of damage that are surrounded by undamaged segments in all directions, including the diagonals) into undamaged ones. Moreover, a dilate morphological operation is performed to minimize discontinuities in damaged areas. In particular, for all the connected regions in the Boolean map (considering 8-connected pixels, i.e., sets of neighboring pixels that touch each other edges or

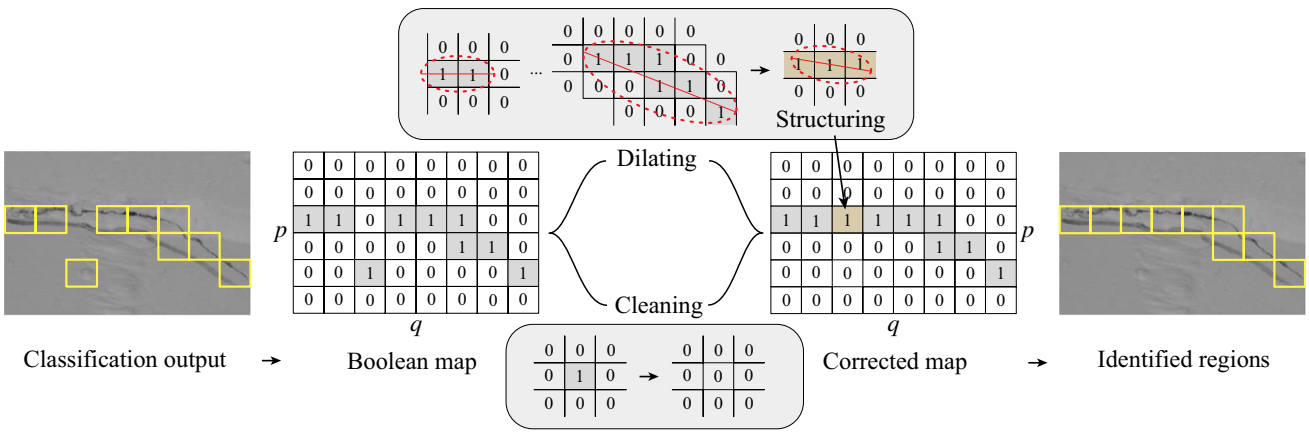


Fig. 3 Morphological operations performed to reduce incorrect classifications

corners), an ellipse having the same second-moment as the analyzed connected region is evaluated. Then, its orientation (i.e., the angle between the horizontal axis and the major axis of the ellipse) is calculated. The dilate operation is, thus, performed using line structuring elements with a length of 3 pixels and an angle given by the average of the orientations calculated throughout the map. An exhaustive theoretical explanation on the dilation and, more generally, on morphological operations is not reported here for the sake of brevity. However, interested readers can find more details in [36].

The final mask that identifies the damaged regions is then obtained by reconverting the corrected Boolean map into a matrix with the same size as the inspection image. The regions selected by this mask can then be processed through Step 2 of the proposed approach to identify the crack pixels. It should be noted that the identified crack regions typically form a small portion of the inspection image. As a result, the number of pixels processed in the next step of the procedure is considerably reduced, decreasing the computational runtime considerably.

2.3 Step 2: crack detection and characterization

In many practical applications, a rough approximation of the crack size is sufficient to prioritize inspections or raising alarms when limited resources are available. Nevertheless, accurate identification of cracked areas, including precise identification and characterization of crack pixels, remains the ultimate goal of crack detection approaches. In this study, only the damaged regions identified in the previous Step 1 are examined to highlight the crack pixels. Step 2 includes a crack detection and a crack characterization sub-step.

Crack detection The first process is applied to localize the crack pixels in the identified damaged regions (Fig. 4a). To this end, the Sobel edge detection method is employed [37]. This method involves calculating the gradient G of a given identified region A as

$$G = \sqrt{G_x + G_y}, \tag{1}$$

where G_x and G_y are two images describing the row and column gradient approximations of A, respectively, calculated as the following two-dimensional convolutions:

$$G_x = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} * A, \tag{2}$$

$$G_y = \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} * A, \tag{3}$$

where $*$ is the convolution operator.

To facilitate further processing, a Boolean map is generated from G by assigning 0 and 1 values to the pixels that are below and above a threshold, which can be defined based on the signal-to-noise ratio (SNR) of the processed image [38] (Fig. 4b). Here, the threshold is automatically set to twice the root mean square (RMS) of G. Two further

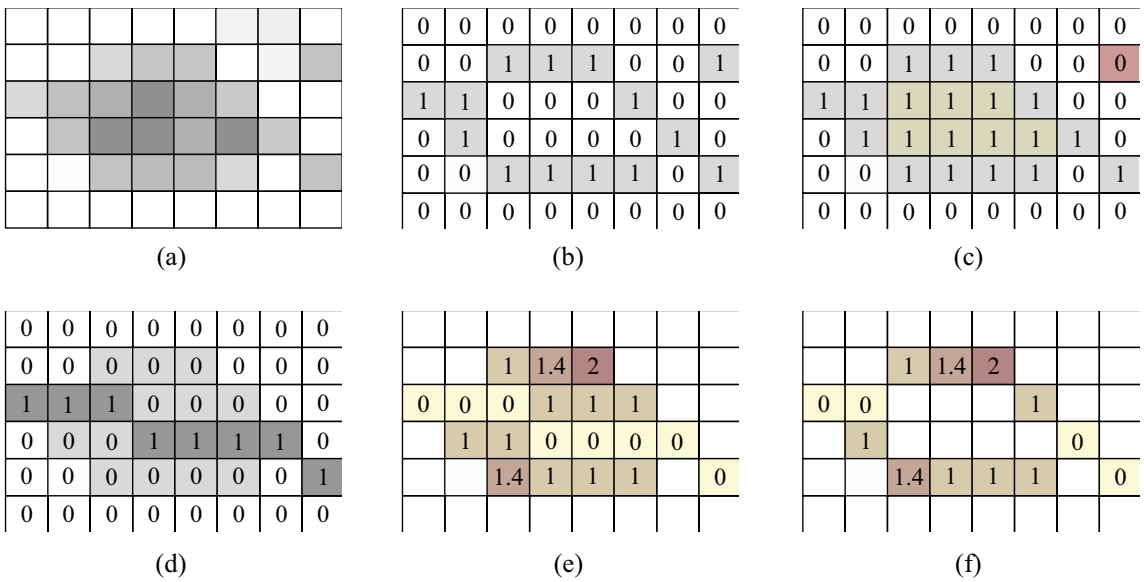


Fig. 4 Crack detection and characterization process: **a** identified damaged region, **b** edge detection, **c** filling and erosion, **d** topological skeleton, **e** distance transform, **f** final distance values

morphological operations are subsequently performed, one consisting of filling, which replaces 0 values with 1 values in closed regions, and the second involves erosion, which removes small objects and smoothens the boundaries of identified areas. Such areas contain the identified crack pixels (Fig. 4c).

Crack characterization Upon localizing the crack pixels, the width and length of cracks are estimated. First, the topological skeleton of the Boolean map obtained from G is calculated using the medial surface axis thinning algorithm [39] (Fig. 4d). Then, the distance transform [40] of the crack pixels is computed, assigning to each pixel a value that represents its distance from the closest point of the identified skeleton, as shown in Fig. 4e. In particular, the Euler distance between the centers of the pixels is employed as distance metric.

The crack length is then calculated as the number of skeleton pixels (i.e., the sum of the pixels in Fig. 4d). On the other hand, a global statistical distribution of the crack width for a given inspection image is calculated as the probability distribution of all the distance values lying on the edge of the identified crack (Fig. 4f). Specifically, the edge values are multiplied by 2 before obtaining the crack distribution, as it is assumed that the crack is locally symmetric with respect to its skeleton. In this study, the crack distribution is obtained by counting how many times each width value is identified in a given figure and representing these results on a width-occurrence diagram. Moreover, a representative value (e.g., the mode or the median of the width statistical distribution) can be calculated to synthetically characterize the crack width.

3 Application

Automatic crack detection is a valuable tool to support and complement traditional visual inspection processes. In this section, the applicability of the method proposed in Sect. 2 is evaluated, with crack detection and characterization examples. The two-step procedure is applied to a real set of photos collected in the proximity of welding joints of a cable-stayed bridge in China [26, 35]. Due to the initial defects of the material and the dynamic load of passing vehicles, fatigue cracks have nucleated around the welding joints over the years. The data employed in this paper were provided in the framework of the 1st International Project Competition for Structural Health Monitoring (ICP-SHM, 2020) [35]. Specifically, a set of 100 randomly selected images (within the set of 120 labeled available images), with a size of 4928×3264 pixels, collected with different camera parameters and lighting conditions, have been used to train the CNN, together with the corresponding labels. A high-resolution consumer-grade camera (Nikon D7000, with a 23.6×15.6 mm CMOS sensor) was used to capture the photos from a distance varying between about 1–2 m. The labels are Boolean pixel maps that have ones and zeros in crack and non-crack areas, respectively. The labels were generated by manually selecting the pixels corresponding to crack areas.

To prepare the data for the training process, each figure is first converted into a single-layer greyscale image and then divided into 32×32-pixel regions. In Fig. 5, three examples of both damaged and undamaged regions are shown for illustration. It can be observed that varying lighting conditions result in significant differences between greyscale samples. Marker notes within the inspection images (Fig. 6a) pose an additional challenge and could affect traditional image processing procedures, such as edge detection, leading to the identification of FPs. The first step of the presented method is intended to mitigate this effect by examining sufficiently small regions, where cracks are clearly distinguishable. In this application, the training process of the CNN reported in Fig. 2 was conducted through 20 epochs using the Adam optimization algorithm [41], with a constant learning rate equal to 0.005, denominator offset 10^{-8}, decay rate of the gradient moving average 0.9, and batch size 128. Training is performed using the MATLAB (R2020b version) software on an Intel® Core™ i7-8700 6@3.20 GHz-processor CPU with 2 GB NVIDIA Quadro P620 GPU, 32 GB RAM, and Windows 10 operating system. The number of undamaged samples in the training process is high (1,553,803) compared to the number of the damaged ones (16,997). However, the same weights are used for both the classes in the loss function to induce robust learning with respect to the wide palette of noise sources in the undamaged regions.

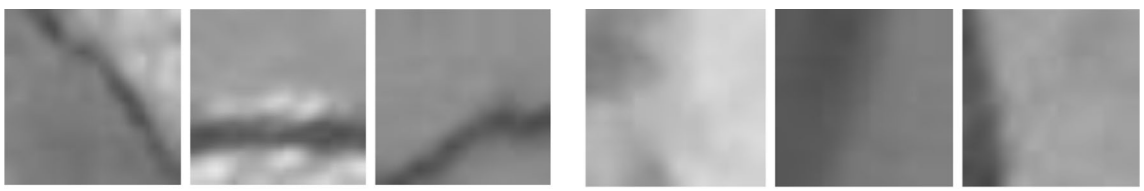


Fig. 5 Examples of damaged (**a**) and undamaged (**b**) regions

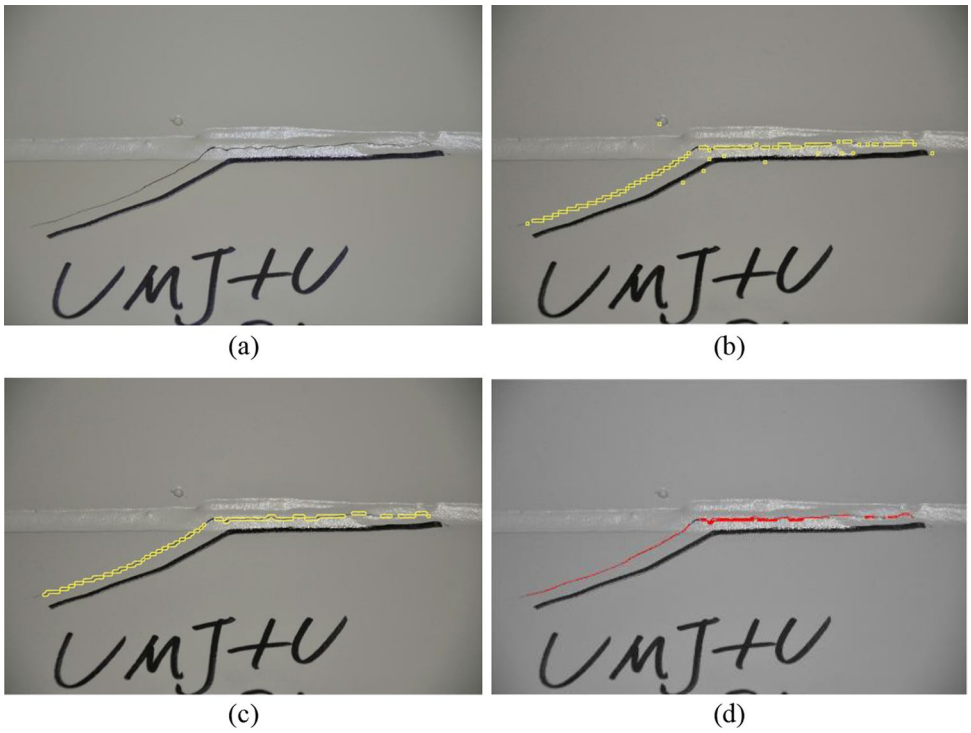


Fig. 6 Crack detection process: **a** original inspection image, **b** Identified crack regions, **c** clean identified crack regions, **d** identified crack pixels

A step-by-step example of the identification procedure is shown in Fig. 6. Specifically, Fig. 6b shows the damaged areas (highlighted with yellow squares) before applying the cleaning process described in Fig. 3. Some FPs and negatives are present due to the uncertainties of the CNN related to changing lighting conditions, the roughness of the surface, and the marker handwrites. After applying the cleaning process, the FPs are entirely removed, while FNs are also reduced, thereby improving the overall quality of the identified regions (Fig. 6c). The identified regions of this last figure are subsequently processed through Step 2, described in Sect. 2.3, resulting in the red crack pixels shown in Fig. 6(d).

Twenty further inspection images, not included in the training set, have been used to test the effectiveness of the proposed procedure.

Step 1 is applied dividing the inspection images into 15,708 samples of size 32×32 pixels. The precision obtained in this step is $p = 98.4\%$—calculated as TPs/(TPs + FPs)—while the recall is $r = 60.8\%$—calculated as TPs/(TPs + FNs)—resulting in an F_1 score of 75.2%—calculated as $2pr/(p + r)$.

As a literature comparison, the results obtained by Xu et al. [26] are reported. The authors of the mentioned paper implemented a fusion convolutional neural network which showed excellent performance with the same dataset employed in this study. Specifically, the authors classified 64×64 images in "crack", "handwriting", and "background". In the mentioned study, considering cracks as "positives" and the other two classes as "negatives", the precision obtained considering 6720 image samples of size 64×64 pixels is 95.5%, while the recall is 93.9%, resulting in an F_1 score of 94.7%. Although the recall obtained in this paper is relatively low compared to the reference study, the simplicity of the CNN and the pixel-level crack identification obtained with a block-level training process are strength points of the presented method. Also, the precision indicator is very high, denoting a correct classification of pristine areas as non-cracked regions, despite the presence of several disturbances.

It should be noted that a higher true-positive rate could be obtained by adjusting the weights of the loss function used in the training procedure, giving more importance to the correct identification of positive samples. However, in this way, the false-positive rate would increase, drawing the attention of expert operators even when positives are caused by disturbance sources. It should be noted that a relatively low value of false negatives does not mean that the cracks are not identified. Indeed, generally, missing positives in Step 1 would only affect the estimated crack length, as shown hereafter.

For the considered inspection images, the crack length and width have been identified using the characterization process presented in Sect. 2.3. The results obtained for each image are reported in Fig. 7. In particular, the crack length shown in Fig. 7a corresponds to the number of pixels identified as a topological skeleton in the identified damaged regions, while the crack width pertains to the mode value of the crack width distribution, which has been identified using the distance transform. These results indicate a general

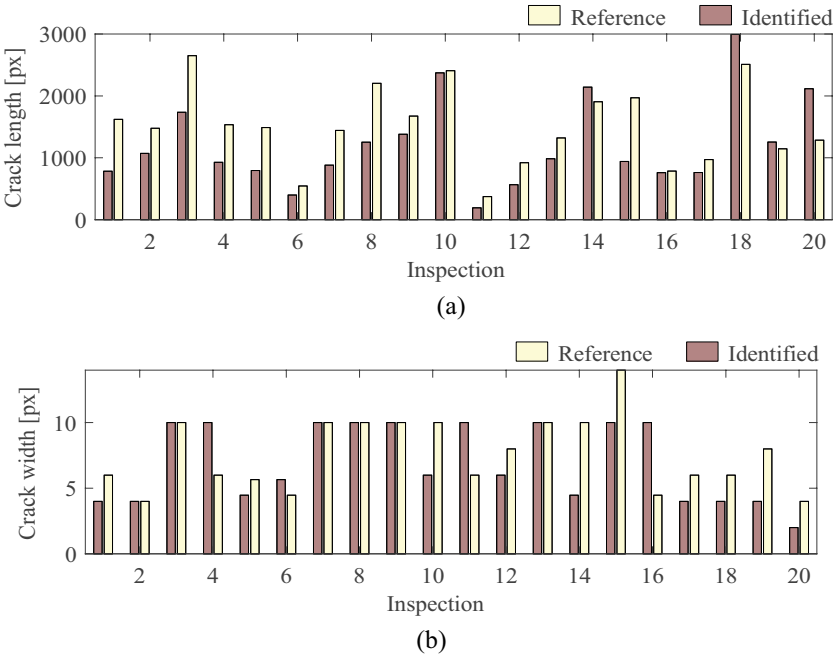


Fig. 7 Identified crack length (**a**) and width (**b**)

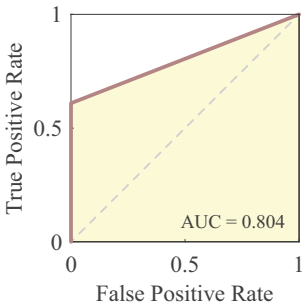


Fig. 8 Receiver operating characteristic curve

underestimation of the crack length, which mainly depends on the presence of FNs in the outcome of the CNN. On the other hand, crack width is typically accurately estimated. Specifically, the average error (in absolute value) on length identification in the 20 inspection images is 31%, while the average width error is 38% (that, however, represents an average error of 2 pixels).

The overall prediction accuracy of Step 1 is illustrated in Fig. 8 using the receiver operating characteristic (ROC) curve that has an area under the curve (AUC) equal to 0.804. For an in-depth analysis of the crack width, the occurrence distributions obtained for every inspection image are reported in Fig. 9, normalized to have a total occurrence value equal to 1. The inspection number is written in the upper-right corner of each plot, in the form I#. In general, the identification results show higher probability values for higher widths when compared against the reference values. This is attributed to noise sources that are incorrectly identified as crack areas during Step 2, as well as to the fact that reference crack pixels are user defined and, thus, may suffer from subjectivity and imprecision in the definition of the crack boundaries. Nevertheless, the mode of the identified occurrence distributions is generally close to that of the reference distributions.

In certain cases, discrepancy is noted in the numerical results shown in Figs. 8 and 9 with respect to the reference values. However, a visual analysis of the identified crack pixels shows that the identification outcome still coincides with actual cracked areas. Figure 10 contains the results obtained for four cases, namely inspections 10, 13, 16, and 18. Inspection 10 has an underestimated crack width (see Fig. 9), inspection 13 results in an underestimated crack length, inspection 16 has a considerably overestimated crack width, while inspection 18 has a slightly overestimated crack length and underestimated crack width using the proposed method.

The original inspection images (left) contain RGB color information, which is removed prior to application of the proposed method. The yellow areas in the central part of Fig. 10 indicate the identified damaged regions after the cleaning process. Some FPs remain, especially in Fig. 10b and d. In the first case, FPs also lead to an overestimation of the crack length. On the other hand, FNs can be found in Fig. 10a and c. However, in these examples, it is possible to notice that marker handwrites of different thicknesses and colors do not affect the efficacy of the crack detection algorithm.

The right-hand side of Fig. 10 shows identified crack pixels superimposed to the manually selected labels. It is

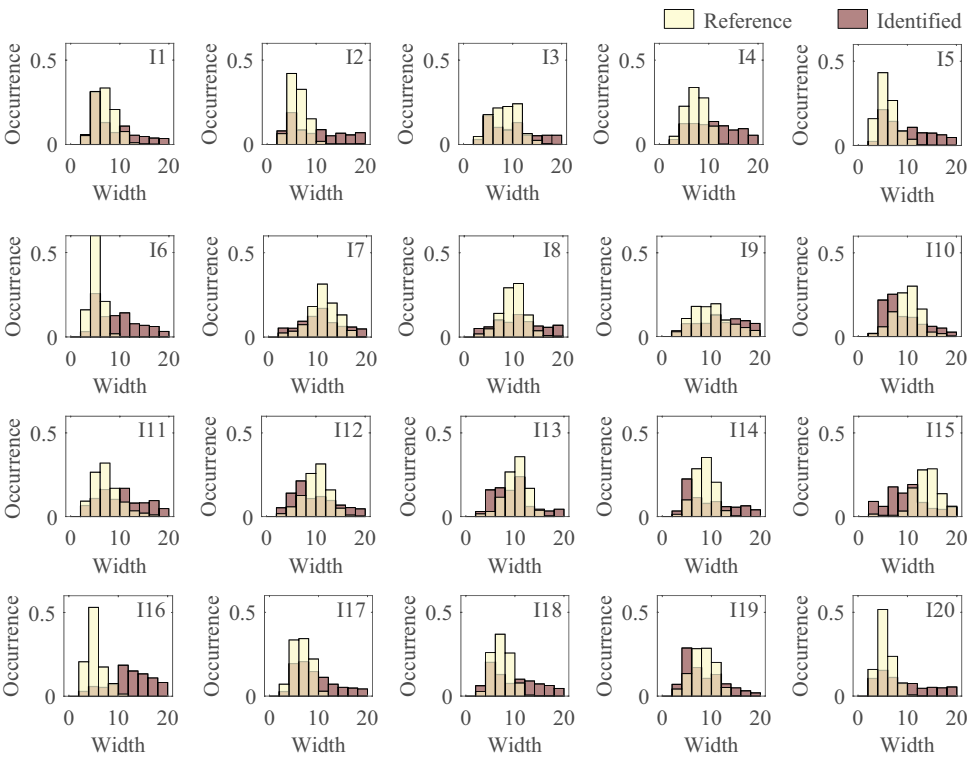


Fig. 9 Crack width distributions

possible to observe that cracked areas are generally well identified, while the identification performance seems independent from camera distances, angles, exposure, and lighting conditions. Indeed, although inspections 16 and 18 have a similar camera angle, the performances in crack size identification differ, as mentioned before. Also, similar results in terms of crack length are obtained for inspections 10 and 13, which are taken under different lighting and exposure conditions.

Figure 11 displays a small detail of Fig. 10a, highlighted as a yellow rectangle in the figure of the identified crack pixels (right). In this figure, it is possible to observe that the identified crack pixels are visually in good agreement with the real crack. Moreover, several bumps and ramifications in the original crack image are well identified (see the area highlighted with dashed contour in Fig. 11). In contrast, the reference label tends to regularize the crack profile and smooth its width. It should be reminded that the labels are obtained by manually (and roughly) selecting the crack areas and, therefore, do not necessarily reflect a real representation of the crack. However, manually selected labels are used in this study to obtain rough estimates of the crack size. The simplification of the reference labels is found to be the main source of discrepancy for the results reported in Figs. 8b and 9.

Using the Nikon D7000 camera employed in this work at a distance of 1–2 m distance and a focal length of 80–100 mm, the minimum detectable width is approximately 0.05 mm/pixel [26]. This limitation dictates a lower boundary in the crack identification size. Indeed, cracks thinner than 0.05 mm would either be identified as cracks with zero-pixel width or not identified. This phenomenon can be observed in the detail highlighted with dashed contour in Fig. 11. The camera employed in this study is considered a consumer-grade camera and is comparable of the cameras employed in most inspection drones employed in literature studies [31].

As a further comparison, the results obtained using only the Sobel edge detection method (applied using the same parameters of Step 2 of the proposed procedure, see Sect. 2.3) are reported in Fig. 12 for the four selected inspection images. Although this method may perform well with modest noise sources, in this case, shadows, surface textures, and handwrite scripts are detected as possible crack areas, highlighting the need for the proposed Step 1.

Further unlabeled inspection images with external objects have been chosen to test the robustness of the procedure with respect to unseen perturbation objects. Figure 13 shows two examples from this dataset that include a red ruler, which was not present in any photo of the training dataset. The identification results are insensitive to this new disturbance source, and the identified crack pixels are in good agreement with the actual crack. Although the numbers and the measurement marks on the rule have a width comparable with that of the cracks and present similar color gradients as cracks, no FPs are detected.

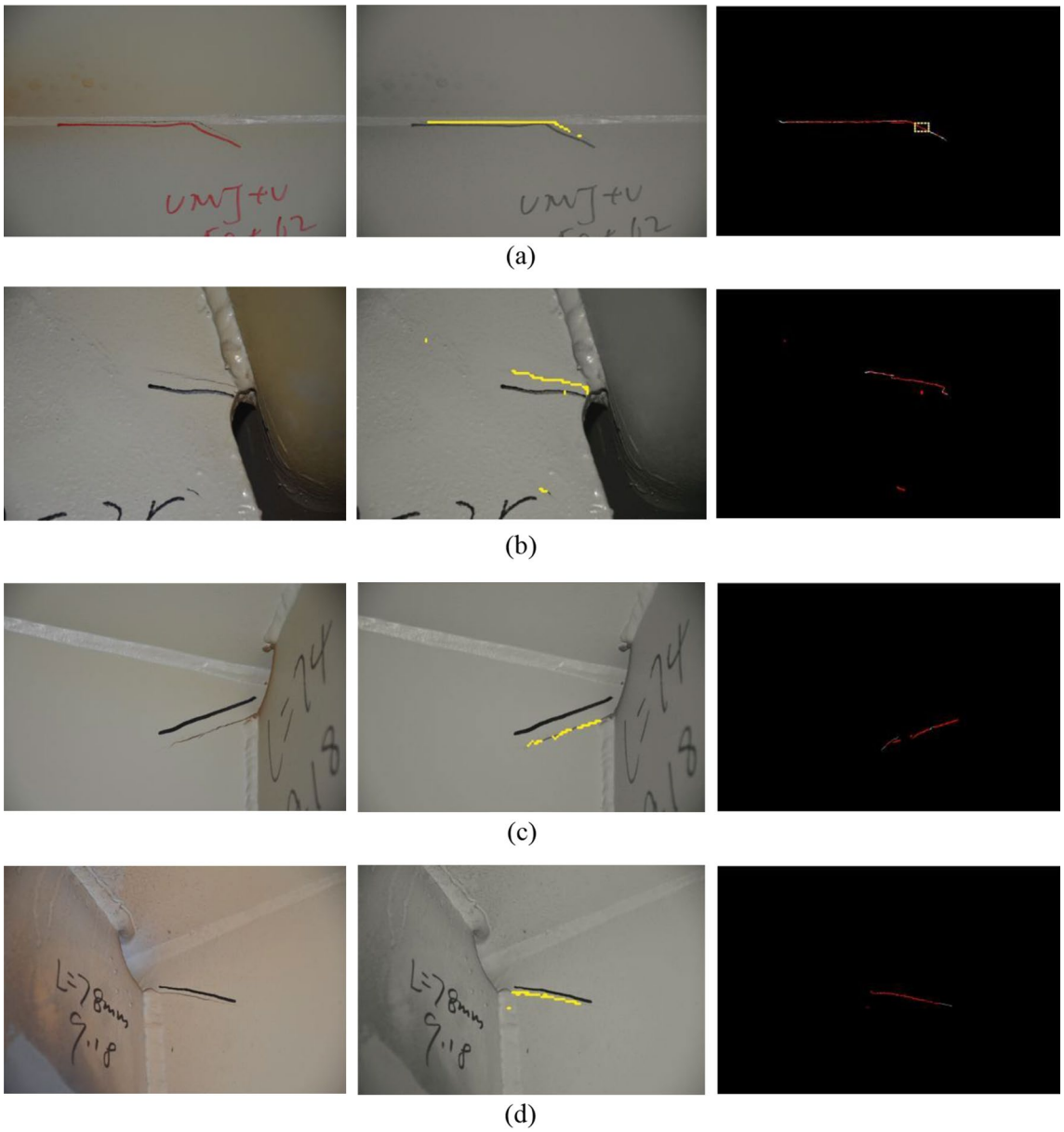
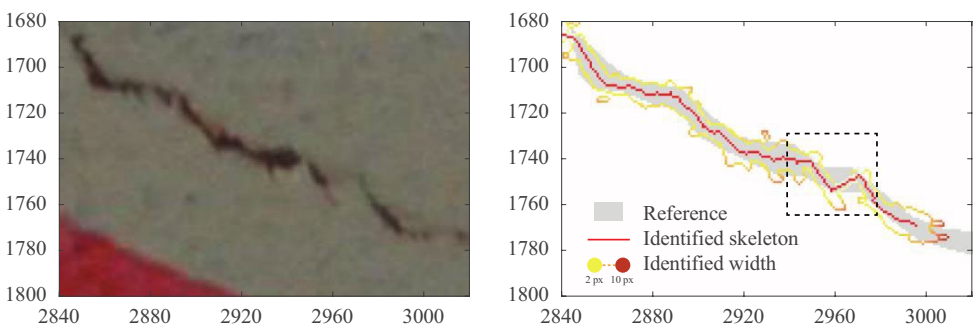


Fig. 10 Crack detection using images from the testing dataset. From the left: original image, identified damaged regions (yellow), and identified crack pixels (red) comparison with labels (white): Inspection 10 (**a**), 13 (**b**), 16 (**c**), and 18 (**d**)

Fig. 11 Detail of the crack pixel detection and characterization process, compared to a reference label



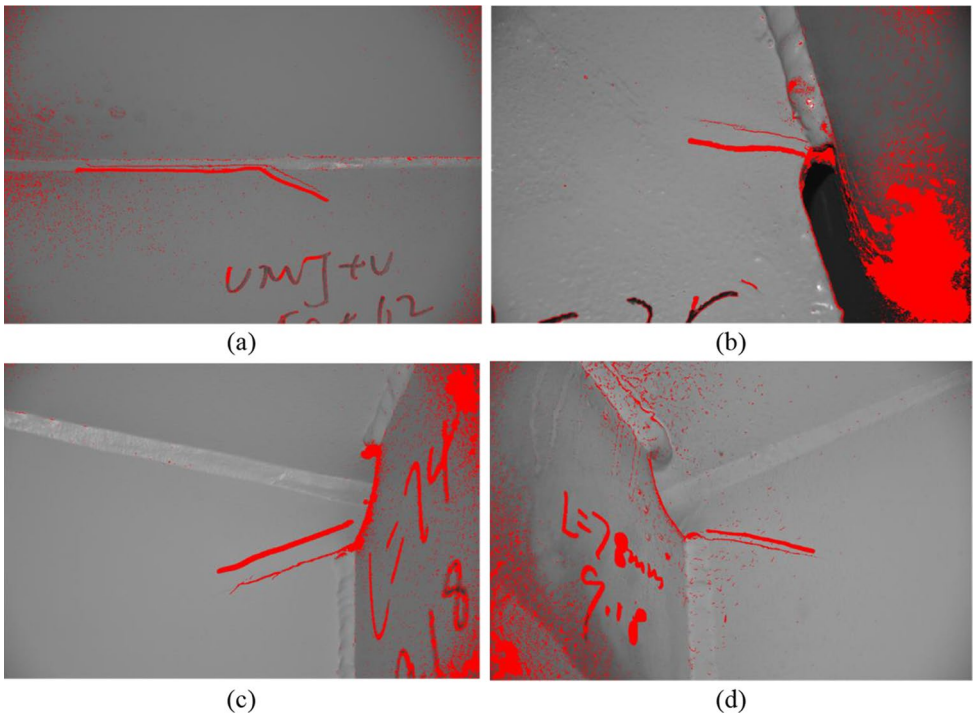


Fig. 12 Results obtained using the Sobel edge detection method: Inspection 10 (**a**), 13 (**b**), 16 (**c**), and 18 (**d**)

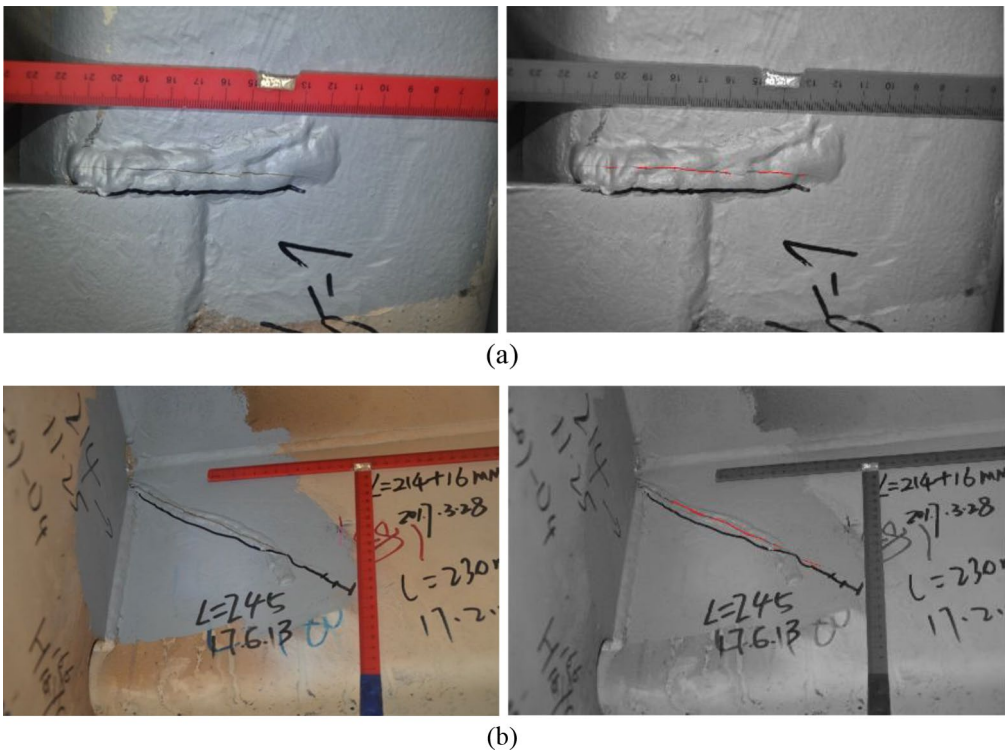


Fig. 13 Crack detection using unlabeled images with external objects; from left: original image and identified crack: Inspection 21 (**a**) and 22 (**b**)

4 Conclusions

In this paper, a method based on convolutional neural networks and edge detection is proposed to identify fatigue cracks in images acquired from welded connection details of steel structures. The method consists of two steps, first, the classification of small regions in the inspection figure using a convolutional neural network and then subsequent identification of crack pixels in the identified damaged areas. To increase the performance in terms of accuracy, a correction procedure based on morphological operations is implemented. At the end of Step 1, an AUC equal to 0.804 is achieved for the testing dataset. The crack characteristics (length and width) identified in the second step are generally in good agreement with the reference values, identified using manually defined crack labels. Although the identified crack pixels and labels present some discrepancies, mainly due to the subjective labeling in the reference images, the results of the proposed identification method are promising and have shown to be robust to different noise sources, as well as the presence of external objects (i.e., a red measurement ruler) that were not part of the training set.

One of the main advantages of the proposed method over existing techniques is the simple structure of the neural network. Moreover, the classification based on small regions allows considerable simplification of the process to generate the training dataset. In the technique presented in this paper, it is not strictly necessary to precisely label crack pixels in the inspection image; a rough selection of damaged regions is sufficient. The proposed two-step procedure is a valuable tool that may help operators to limit the subjectivity of their evaluations and automatize the crack detection process. In addition, the simple structure of the neural net reduces the required computation power and thus, may allow for computer-aided in situ inspection schemes.

Acknowledgements The authors would like to kindly acknowledge the organizers of the 1st International Project Competition for Structural Health Monitoring (IPC-SHM 2020), ANCRiSST, Harbin Institute of Technology (China), and University of Illinois at Urbana-Champaign (USA) for their generously providing the invaluable data from actual structures. The authors also would like to thank the chairs of IPC-SHM 2020 Prof. Hui Li, and Prof. Billie F. Spencer Jr for their leadership on the competition. Research described in this paper was financially supported by the Real-time Earthquake Risk Reduction for a Resilient Europe 'RISE' project, financed under the European Union's Horizon 2020 research and innovation programme, under grant agreement No 821115, by the ETH Grant (ETH-11 18-1) Dynarisk—"Enabling Dynamic Earthquake Risk Assessment", as well as by the Singapore-ETH center (SEC) under contract no. FI 370074011-370074016.

Funding Research described in this paper was financially supported by the Real-time Earthquake Risk Reduction for a Resilient Europe 'RISE' project, financed under the European Union's Horizon 2020 research and innovation programme, under grant agreement No 821115, by the ETH Grant (ETH-11 18-1) Dynarisk—"Enabling Dynamic Earthquake Risk Assessment", as well as by the Singapore-ETH center (SEC) under contract no. FI 370074011-370074016.

Availability of data and materials The data supporting the results reported in this paper have been provided by the organizers of the 1st International Project Competition for Structural Health Monitoring (IPC-SHM 2020).

Code availability Code is available upon motivated request.

Declarations

Conflict of interest The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

1. MacDougall C, Green MF, Shillinglaw S (2006) Fatigue damage of steel bridges due to dynamic vehicle loads. J Bridg Eng 11:320–328. https://doi.org/10.1061/(ASCE)1084-0702(2006)11:3(320)
2. Farrar CR, Worden K (2012) Structural health monitoring. Wiley, Chichester
3. Brincker R, Ventura CE (2015) Introduction to operational modal analysis. Wiley, Chichester
4. Brownjohn JMW, De Stefano A, Xu Y-L et al (2011) Vibration-based monitoring of civil infrastructure: challenges and successes. J Civ Struct Heal Monit 1:79–95. https://doi.org/10.1007/s13349-011-0009-5
5. Bao Y, Li H (2020) Machine learning paradigm for structural health monitoring. Struct Heal Monit. https://doi.org/10.1177/1475921720972416
6. Lynch JP (2006) A summary review of wireless sensors and sensor networks for structural health monitoring. Shock Vib Dig 38:91–128. https://doi.org/10.1177/0583102406061499
7. Jang S, Jo H, Cho S et al (2010) Structural health monitoring of a cable-stayed bridge using smart sensor technology: deployment and evaluation. Smart Struct Syst 6:439–459. https://doi.org/10.12989/sss.2010.6.5_6.439
8. Klis R, Chatzi EN (2017) Vibration monitoring via spectro-temporal compressive sensing for wireless sensor networks. Struct Infrastruct Eng 13:195–209. https://doi.org/10.1080/15732479.2016.1198395
9. Chen FC, Jahanshahi MR (2018) NB-CNN: deep learning-based crack detection using convolutional neural network and naïve bayes data fusion. IEEE Trans Ind Electron 65:4392–4400. https://doi.org/10.1109/TIE.2017.2764844
10. Cha YJ, Choi W, Suh G et al (2018) Autonomous structural visual inspection using region-based deep learning for detecting multiple damage types. Comput Civ Infrastruct Eng 33:731–747. https://doi.org/10.1111/mice.12334
11. Cha YJ, Choi W, Büyüköztürk O (2017) Deep learning-based crack damage detection using convolutional neural networks. Comput Civ Infrastruct Eng 32:361–378. https://doi.org/10.1111/mice.12263
12. Yeum CM, Dyke SJ (2015) Vision-based automated crack detection for bridge inspection. Comput Civ Infrastruct Eng 30:759–770. https://doi.org/10.1111/mice.12141
13. Kim B, Cho S (2018) Automated vision-based detection of cracks on concrete surfaces using a deep learning technique. Sensors (Switz). https://doi.org/10.3390/s18103452

14. Bae H, Jang K, An Y-K (2020) Deep super resolution crack network (SrcNet) for improving computer vision–based automated crack detectability in in situ bridges. Struct Heal Monit. https://doi.org/10.1177/1475921720917227
15. Abdel-Qader I, Abudayyeh O, Kelly ME (2003) Analysis of edge-detection techniques for crack identification in bridges. J Comput Civ Eng 17:255–263. https://doi.org/10.1061/(ASCE)0887-3801(2003)17:4(255)
16. Nishikawa T, Yoshida J, Sugiyama T, Fujino Y (2012) Concrete crack detection by multiple sequential image filtering. Comput Civ Infrastruct Eng 27:29–47. https://doi.org/10.1111/j.1467-8667.2011.00716.x
17. Alaknanda ARS, Kumar P (2009) Flaw detection in radiographic weldment images using morphological watershed segmentation technique. NDT E Int 42:2–8. https://doi.org/10.1016/j.ndteint.2008.06.005
18. Cha YJ, Chen JG, Büyüköztürk O (2017) Output-only computer vision based damage detection using phase-based optical flow and unscented Kalman filters. Eng Struct 132:300–313. https://doi.org/10.1016/j.engstruct.2016.11.038
19. Mohan A, Poobal S (2018) Crack detection using image processing: a critical review and analysis. Alexandria Eng J 57:787–798. https://doi.org/10.1016/j.aej.2017.01.020
20. Steinkraus D, Buck I, Simard PY (2005) Using GPUs for machine learning algorithms. Proc Int Conf Doc Anal Recogn ICDAR 2005:1115–1120. https://doi.org/10.1109/ICDAR.2005.251
21. Butcher JB, Day CR, Austin JC et al (2014) Defect detection in reinforced concrete using random neural architectures. Comput Civ Infrastruct Eng 29:191–207. https://doi.org/10.1111/mice.12039
22. Huyan J, Li W, Tighe S et al (2020) CrackU-net: a novel deep convolutional neural network for pixelwise pavement crack detection. Struct Control Heal Monit. https://doi.org/10.1002/stc.2551
23. Li S, Zhao X (2019) Image-based concrete crack detection using convolutional neural network and exhaustive search technique. Adv Civ Eng 2019:1–12. https://doi.org/10.1155/2019/6520620
24. Zhang L, Yang F, Daniel Zhang Y, Zhu YJ (2016) Road crack detection using deep convolutional neural network. In: 2016 IEEE International conference on image processing (ICIP). IEEE, pp 3708–3712
25. Dung CV, Anh LD (2019) Autonomous concrete crack detection using deep fully convolutional neural network. Autom Constr 99:52–58. https://doi.org/10.1016/j.autcon.2018.11.028
26. Xu Y, Bao Y, Chen J et al (2019) Surface fatigue crack identification in steel box girder of bridges by a deep fusion convolutional neural network based on consumer-grade camera images. Struct Heal Monit 18:653–674. https://doi.org/10.1177/1475921718764873
27. Sathya K, Sangavi D, Sridharshini P et al (2020) Improved image based super resolution and concrete crack prediction using pretrained deep learning models. Soft Comput Civ Eng 4:40–51. https://doi.org/10.22115/SCCE.2020.229355.1219
28. Jin S, Lee SE, Hong J-W (2020) A vision-based approach for autonomous crack width measurement with flexible kernel. Autom Constr 110:103019. https://doi.org/10.1016/j.autcon.2019.103019
29. Qiao W, Ma B, Liu Q et al (2021) Computer vision-based bridge damage detection using deep convolutional networks with expectation maximum attention module. Sensors 21:824. https://doi.org/10.3390/s21030824
30. Bhowmick S, Nagarajaiah S, Veeraraghavan A (2020) Vision and deep learning-based algorithms to detect and quantify cracks on concrete surfaces from UAV videos. Sensors 20:6299. https://doi.org/10.3390/s20216299
31. Zollini S, Alicandro M, Dominici D et al (2020) UAV photogrammetry for concrete bridge inspection using object-based image analysis (OBIA). Remote Sens 12:3180. https://doi.org/10.3390/rs12193180
32. Ellenberg A, Branco L, Krick A et al (2015) Use of unmanned aerial vehicle for quantitative infrastructure evaluation. J Infrastruct Syst 21:04014054. https://doi.org/10.1061/(ASCE)IS.1943-555X.0000246
33. Sun X, Jiang Y, Ji Y et al (2019) Distance measurement system based on binocular stereo vision. IOP Conf Ser Earth Environ Sci 252:052051. https://doi.org/10.1088/1755-1315/252/5/052051
34. Pohl D, Dorodnicov S, Achtelik M (2019) Depth map improvements for stereo-based depth cameras on drones, pp 341–348
35. Bao Y, Li J, Nagayama T et al (2021) The 1st international project competition for structural health monitoring (IPC-SHM, 2020): a summary and benchmark problem. Struct Heal Monit. https://doi.org/10.1177/14759217211006485
36. Soille P (2004) Morphological image analysis. Springer, Berlin Heidelberg
37. Kanopoulos N, Vasanthavada N, Baker RL (1988) Design of an image edge detection filter using the Sobel operator. IEEE J Solid-State Circuits 23:358–367. https://doi.org/10.1109/4.996
38. Pratt WK (2006) Edge detection. In: Digital image processing, Fourth edition. Wiley, Hoboken, pp 465–533
39. Lee TC, Kashyap RL, Chu CN (1994) Building skeleton models via 3-D medial surface axis thinning algorithms. CVGIP Graph Model Image Process 56:462–478. https://doi.org/10.1006/cgip.1994.1042
40. Maurer CR, Qi R, Raghavan V (2003) A linear time algorithm for computing exact Euclidean distance transforms of binary images in arbitrary dimensions. IEEE Trans Pattern Anal Mach Intell 25:265–270. https://doi.org/10.1109/TPAMI.2003.1177156
41. Kingma DP, Ba JL (2015) Adam: a method for stochastic optimization. In: 3rd international conference on learning representations, ICLR 2015—conference track proceedings